REMARKS

An Excess Claim Fee Payment Letter is submitted herewith to cover the cost of two (2) excess total claims.

Claims 1-22 are all the claims presently pending in the application. Claims 1, 4, 6-9, 11-13, 15, 17 and 20 have been amended to more particularly define the invention. Claims 21-22have been added to claim additional features of the invention.

It is noted that the claim amendments are made only for more particularly pointing out the invention, and not for distinguishing the invention over the prior art, narrowing the claims or for any statutory requirements of patentability. Further, Applicant specifically states that no amendment to any claim herein should be construed as a disclaimer of any interest in or right to an equivalent of any element or feature of the amended claim.

Claims 6 and 7 stand rejected under 35 U.S.C. §112, second paragraph as allegedly being indefinite.

Claims 1-13 and 17-20 stand rejected under 35 U.S.C. §102(b) as allegedly being anticipated by Mellor-Crummey ("Compile-time Support for Efficient Data Race Detection in Shared-Memory Parallel Programs"). Claims 14-16 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Mellor-Crummey in view of Flanagan et al. (U. S. Patent No. 6,343,371).

These rejections are respectfully traversed in the following discussion.

I. THE CLAIMED INVENTION

The claimed invention (e.g., as recited in claim 1 and similarly recited in claims 17 and 20) is directed to a method for statically detecting a datarace condition in a multithreaded application. The method includes inputting a set of input information, processing the set of input information by comparing threads that may execute statements in a statement pair, and outputting a statement conflict set that identifies the statement pairs whose execution instances definitely or potentially cause dataraces, without executing the multithreaded application.

Conventional methods (e.g., static datarace detection methods) identify many dataraces that may never be exhibited in any program execution (e.g,."false positives"). In addition, programmer annotations are required to improve the effectiveness of such conventional tools (Application at page 6, lines 1-10).

The claimed invention, on the other hand, processes the set of input information by comparing threads that may execute statements in a statement pair (Application at Figure 3; page 8, line 18-page 9, line 9; page 14, lines 9-12). This feature allows the claimed method (e.g., and system) to more accurately detect a datarace condition (e.g., a condition where a datarace definitely will occur or may possibly occur) than in conventional methods (Application at page 6, lines 12-15).

II. THE 35 USC 112, SECOND PARAGRAPH REJECTION

The Examiner alleges that claims 6-7 are indefinite. Applicant notes, however, that these claims have been amended to depend from claim 5, to address the Examiner's concerns. Therefore, these claims are not indefinite.

In view of the foregoing, the Examiner is respectfully requested to withdraw this rejection.

III. PRIOR ART REFERENCES

A. The Mellor-Crummey Reference

The Examiner alleges that Mellor-Crummey teaches the claimed invention of claims 1-13 and 17-20. Applicant submits, however, that there are elements of the claimed invention which are neither taught nor suggested by Mellor-Crummey.

Mellor-Crummey discloses a compile-time support system for datarace detection in shared-memory programs. Specifically, Mellor-Crummey discloses ERASER, a datarace instrumentation tool that uses program analysis to prune the number of references to be monitored (Mellor Crummey at Abstract).

However, Mellor-Crummey does not teach or suggest "*processing the set of input*

information by comparing threads that may execute statements in a statement pair", as recited, for example, in claim 1 and similarly recited in claims 17 and 20.

As noted above, unlike conventional methods (e.g., static datarace detection methods) which identify many dataraces that may never be exhibited in any program execution (e.g,."false positives," the claimed invention processes the set of input information by comparing threads that may execute statements in a statement pair (Application at Figure 3; page 8, line 18-page 9, line 9; page 14, lines 9-12). This feature allows the claimed method (e.g., and system) to more accurately detect a datarace condition (e.g., a condition where a datarace definitely will occur or may possibly occur) than in conventional methods (Application at page 6, lines 12-15).

Clearly, this feature is not taught or suggested by Mellor Crummey. Indeed, Mellor-Crummey is completely unrelated to the claimed invention.

For example, Applicant would point out that Mellor-Crummey's ParaScope is intended for nested fork/join or cobegin/coend style parallelism where "threads" creations/terminations are all well-nested.

The claimed invention, on the other hand, may involve general multithreaded applications such as Java, C or C++. Thus, **the static analysis of an exemplary aspect of the claimed invention may explicitly tag each statement with the set of threads that may execute it and compare these sets of different statements to determine whether two statements can cause a datarace during execution** (e.g., see Application at page 15, line 8-page 18, line 12; Figures 3, 4 and 6) Nowhere is this feature taught or suggested by Mellor-Crummey.

Further, in general multithreaded applications, accesses to a memory location by different threads can be controlled by locks. Even assuming (arguendo) that ERASER may use this information, Mellor-Crummey's work clearly does not include this analysis. Again, as noted above, in an exemplary aspect of the claimed invention, **static analysis may compare the sets of locks held by threads while executing statements in deciding whether the executions may cause (e.g., definitely or potentially cause) a datarace during execution**.

The Examiner attempts to rely on page 132, right column in Mellor-Crummey to support his position. However, this passage merely states that "[d]ata dependence analysis is a deep

compile-time analysis of program variables and their subscripts to determine when two variable references may refer to the same memory location". That is, this passage merely refers to "data dependence analysis". Nowhere does this passage teach or suggest processing the set of input information by comparing threads that may execute statements in a statement pair, as in the exemplary aspects of the claimed invention.

Therefore, Applicant respectfully submits that Mellor-Crummey does not teach or suggest each and every element of the claimed invention. Therefore, the Examiner is respectfully requested to withdraw this rejection.

B. The Flanagan Reference

The Examiner alleges that Mellor-Crummey would have been combined with Flanagan to form the invention of claims 14-16. Applicant submits, however, that these references would not have been combined and even if combined, the alleged combination would not teach or suggested each and every element of the claimed invention.

Flanagan discloses a system for statically detecting potential race conditions in a multi-threaded computer program. In the system, a race condition detector determines whether accesses to object data fields are consistently protected by an appropriate lock. An object data field access that is not protected by an appropriate lock indicates a potential race condition (Flanagan at Abstract).

Applicant respectfully submits that these references would not have been combined as alleged by the Examiner. Indeed, these references are completely unrelated, and no person of ordinary skill in the art would have considered combining these disparate references, absent impermissible hindsight.

In fact, Applicant submits that these references do not include any motivation or suggestion to urge the combination as alleged by the Examiner. Indeed, these references clearly do not teach or suggest their combination.

Therefore, Applicant respectfully submits that one of ordinary skill in the art would not have been so motivated to combine the references as alleged by the Examiner. Therefore, the

Examiner has failed to make a prima facie case of obviousness.

Moreover, Applicant submits that neither Mellor-Crummey, nor Flanagan, nor any alleged combination thereof, teaches or suggests "*processing the set of input information by comparing threads that may execute statements in a statement pair*", as recited, for example, in claim 1 and similarly recited in claims 17 and 20. As noted above, this feature allows the claimed method (e.g., and system) to more accurately detect a datarace condition (e.g., a condition where a datarace definitely will occur or may possibly occur) than in conventional methods (Application at page 6, lines 12-15).

Clearly, this feature is not taught or suggested by Flanagan. Indeed, Flanagan is clearly unrelated to the claimed invention. In fact, the Examiner is not even attempting to rely on Flanagan as teaching this feature, but merely relies on Flanagan as allegedly teaching an object-oriented programming language.

Further, Flanagan makes clear that he identifies a potential race condition, not by comparing threads, but by determining whether an object data field access is not protected by an appropriate lock (e.g., see Flanagan at col. 7, line 64-page 8, line 27). Thus, Flanagan is clearly unrelated to the claimed invention, in which the set of input information is processed by comparing threads that may execute statements in a statement pair.

Therefore, Applicant submits that these references would not have been combined and even if combined, the combination would not teach or suggest each and every element of the claimed invention. Therefore, the Examiner is respectfully requested to withdraw this rejection.

III. FORMAL MATTERS AND CONCLUSION

In view of the foregoing, Applicant submits that claims 1-22, all the claims presently pending in the application, are patentably distinct over the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to

discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Assignee's Deposit Account No. 50-0510.

Respectfully Submitted,

Date: 2/24/05

Phillip E. Miller, Esq.
Registration No. 46,060

McGinn & Gibb, PLLC
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254